SYNTAX ZX80®

A PUBLICATION OF **THE HARVARD GROUP**

IN THIS ISSUE

INDEX OF ADVERTISERS

SINCLAIR DISCONTINUES TECH INFO LINE

Nigel Searle of Sinclair just announced that Sinclair would no longer staff their technical information line in Boston. For details, see his letter, p.10.

EDGE CONNECTORS AVAILABLE

Kopak Creations of New York now offers 46-pin edge connectors for ZX80/81s. Robert Schiller of Kopak says PVC connectors are available now for $15. You'll get solder tail and wire wrap connectors for about the same price soon. Kopak Creations, 448 W. 55th St, New York, NY, 10019, 212/757-8698.

NONAUTHORIZED ZX81 SALES AND SERVICE

National Electronics III of Concord, TN, has become a nonauthorized reseller of new assembled ZX81 systems. Cost: $149.95 plus $4.95 shipping and handling.
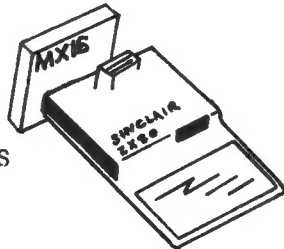According to Jeffrey Thompson, owner of National, they also perform out-of-warranty repairs on ZX81s, charging actual labor time ($10/hour, $10 minimum) plus parts. He says most repairs take 1 hour or less. Their turn-around time is about 48 hours. For service, send your computer to National with a description of the problem. They will write or call you collect with an estimate. Contact National Electronics III, 10425 Kingston Pike, Concord, TN, 37922, 615/ 690-8927, 10-6 M-F, 10-4 Sat.

SYNTAX BINDERS

Now you don't have to wait to renew to get a SYNTAX binder to protect your issues. You can get the magazine binders we send to early renewals for $9.00, including shipping. These sturdy binders hold 14 issues with wires--no punching necessary. SYNTAX, RD 2 Box 457, Harvard, MA 01451.

SYNTAX ERRORS; Peter Ulrich, New York, sent these corrections for Paul Ezra's Brownian Motion program (Dec.81). He explains that Paul's program runs, but gives curved lines because the manual's subroutine uses N for 2 different variables. Change the subroutine:
1150 FOR L=0 TO M
1250 NEXT L
This allows these program changes:
   40 LET D=INT (RND*42)+1
   50 LET A=C
   60 LET B=D

David Ornstein corrected his New 8K ROM Routine Addresses list (Jan.82) issue: LIST at 072C should be LLIST (LIST is at 0730); DCDKB is at 07BD, not 07BC; and RAND is at 0E6C, not 0E69.

In 4K/1K Blackjack (Jan.82 p.16), line 670 should read PRINT P(Z),CHR$(9) to mask the card.

Jon Passler sent some improvements for his 8K Bar Chart program (Jan. 82) to speed up program execution:
   70 IF B(A)=0 THEN STOP
   80 FOR P=1 TO B(A)/100+.5
   90 PLOT A*2+3,P
  100 NEXT P
  110 NEXT A

Final Monzxer changes (Nov.81): Add 310 PLOT 23,32. Line 2120 should read 2120 LET A(1)=YR

6K RAM ADDITION

Frustrated by the limitations of 2K RAM, Jim Graham of Oakham, MA, put 6K "under the hood" of his ZX81. He finds it a comfortable compromise, freeing up the edge connector for I/O use. The wire-wrapped module consists of four sockets welded together, needs no board modifications and plugs into the existing RAM socket. For plans, send $2.00 and SASE. Jim Graham, East Hill, Oakham, MA 01068.

SURPLUS KEYBOARDS PLUS DIRECTIONS

The $20 surplus keyboards from John J. Meshna Jr., Inc., mentioned in the Dec.81 issue, now come with wiring instructions for ZX80s. If you got one without instructions, call Doug Meshna for a free copy. Stock no. SPL-19, John J. Meshna Jr., Inc., PO Box 62, E. Lynn, MA, 01904, 617/595-2275.

DISCOUNT ICS

Steve Streebing offers SYNTAX readers factory prime RAM ICs at low rates: HIT 6116-3 16K (2Kx8) CMOS RAM (150nS) for $14.50, 2114L-2 4K (1Kx4) static RAM (200nS) for $3.25 each or 16 for $48.00 (US funds, add $1.25 for shipping). Al Salt used 2114s in his Build Additional RAM (Mar.81) and others use 6116s as plug-in replacements for the 1K RAM on-board (4118) in ZX81s with 28-pin sockets. Write Steve at Box 182, Washington, MI, 48094.

2

TRANSLATING BASICS--WHAT WON'T WORK

With all the software available for other computers, many ZX80 owners want to convert programs from Microsoft BASIC (used in PETs, Apples, and TRS-80s) to ZX80 BASIC. Sometimes this proves futile due to inconsistencies between Microsoft and ZX80 BASIC.

To know if your attempts to convert to ZX80 BASIC might work, here are some Microsoft BASIC keywords that cannot be practically translated.

Keyword	Description
INT(x)	computes the integer value of x; 4K BASIC does this automatically
HCOLOR=x	used by Apple to set high resolution graphics color
HPLOT SET RESET DRAW XDRAW PLOT	all used to light up a graphics block (pixel) or a set of pixels
SYS x	executes an assembly language subroutine with starting address x. 4K USR(x) does the same job, but since SYS works on computers with a 6502 processor, memory locations will be incompatible
*GETx$	used to receive 1 char. from keyboard and store it as x$
ON ERR GO TO x	sends computer to line x if an error occurs
OPEN CLOSE CMD x PR#x IN#x GET #x INPUT #x PRINT #x	all are used for I/O control; since the 4K machine does not create files, it has no functions like these

*can be converted, but requires too much memory to be practical.

Richard Forsen, New Hartford, NY

PHONE NUMBER WORD GENERATOR--4K-8K

What does your phone number spell out if changed to letters? A friend's phone number, 438-5678, spells GET LOST. A bank's auto loan department answers NEXT CAR.

To encode your number into a readable word, you could just gaze at your telephone dial. But every digit has three letters. A seven digit phone number has 2,187 different possibilities! Now a ZX80/1 (4K/1K, 8K/2K) can do it for you.

The program initializes two counters, P and Q (lines 5&7). P counts combinations printed 40 at a time. Q keeps a running total till the ZX80 prints 2187 combinations. You enter the digits via lines 10-22. The 2187 words are produced by 2187 iterations through the seven nested FOR loops (lines 24-110). The subroutine at 500-506 prints each letter by converting the digit into a printable ASCII value. It converts 2, for example, into ASCII values 38, 39, and 40, which print A, B, and C. The subroutine at 300-306 signals the ZX80 to print the next 40 combinations. Lines 200-202 end the program.

RUN, then enter your phone number (hit NL after each digit). About 15 seconds later the first 40 combinations appear. Press NL to see successive groups. The last one ends with "FINE." Don't input the digits 1 and 0--they have no letters on the phone dial. The program suppresses Q and Z since they do not appear on the dial.

This program was inspired by Harvey M. Deitel's Introduction to Computer Programming (Prentice-Hall), a fine book for learning BASIC.

Lawrence Souder, Philadelphia, PA

```
  5 LET P=0
  7 LET Q=0
 10 INPUT A
 12 INPUT B
 14 INPUT C
 16 INPUT D
 18 INPUT E
 20 INPUT F
 22 INPUT G
 24 FOR Z=0 TO 2
 26 FOR Y=0 TO 2
 28 FOR X=0 TO 2
 30 FOR W=0 TO 2
 32 FOR V=0 TO 2
 34 FOR U=0 TO 2
 36 FOR T=0 TO 2
 38 LET M=A
 40 LET N=T
 42 GO SUB 500
 44 LET M=B
 46 LET N=U
 48 GO SUB 500
 50 LET M=C
 54 LET N=V
 56 GO SUB 500
 58 LET M=D
 60 LET N=W
 62 GO SUB 500
 64 LET M=E
 66 LET N=X
 68 GO SUB 500
 70 LET M=F
 72 LET N=Y
 74 GO SUB 500
 76 LET M=G
 78 LET N=Z
 80 GO SUB 500
 82 PRINT " ";
 84 LET P=P+1
 85 LET Q=Q+1
 86 IF P>39 THEN GO SUB 300
 88 IF Q=2187 THEN GO TO 200
 90 NEXT T
 92 NEXT U
 94 NEXT V
 96 NEXT W
 98 NEXT X
100 NEXT Y
110 NEXT X
200 PRINT "FINE"
202 STOP
300 INPUT R$
302 LET P=0
304 CLS
306 RETURN
500 LET J=2*M+32+M+N
502 IF J>53 THEN LET J=J+1
504 PRINT CHR$(J);
506 RETURN
```

Syntactic Sum:-29058, 4K; 44104, 8K

BINARY ADDITION AND SUBTRACTION

(This is the second part of this intro to machine code programming series. See also last month's column and Computer Number Systems, Feb.81 p.10, for more on binary numbers.--AZ)

The point to learning how to add and subtract binary numbers is to see how "carries" are generated and what happens to those extra digits.

In binary, answers to arithmetic operations are the same as in decimal; they just look different. For example, in decimal, 1+1=2. In binary, 1+1=10 (or 2 in decimal). The first digit, 1, is the carry. Addition is straightforward; just remember that 0+0=0, 1+0=1, 1+1=10 (or 0 and carry), 1+1+carry=1 and carry:

```
  10001101 = 141    00110101 =  53
 +00110010 =  50   +01011011 =  91
  10111111 = 191    10010000 = 144
```

The largest number you can represent in binary with 8 bits is 255. To get an answer greater than 255, the Z80 (the central processor, or CPU, in your ZX80/81) has a 9th bit available to the Accumulator (where all bit manipulation takes place): the Carry bit in the Flags register. Disregarding just how the carry is transferred to the high order byte, this is generally how large numbers are saved in 2 bytes or addresses:

```
HIGH ORDER (2nd)       LOW ORDER (1st)
  00000000               10010111 = 151
 +00000001 ← Carry ←    +10101000 = 168
  00000001 = 256         00111111 =  63
     256+63 = 319
```

The low order byte is the first of the two addresses; the high order byte is the second. See last month's column for details.
Subtraction is a little trickier. Let's subtract 45 from

152: First, the Z80 "inverts" 45, or changes all the zeros to ones and ones to zeros (00101101--. 11010010). It then adds 1, giving 11010011, which is 211 in decimal. This is the two's complement of 45. This conversion to two's complement explains why if you POKE -45, then PEEK that location, you get 211.
Next, the Z80 adds 211 to 152:

```
  10011000 = 152
 +11010011 = -45 or 211
  01101011 = 107 (ignore the carry)
```

The correct answer is 107.
Depending on the results of addition and "subtraction" (conversion to two's complement followed by addition), certain flags in the Accumulator-Flags "register-pair" will be changed.

NEXT MONTH: THE FLAGS REGISTER

Jon Bobst, Zeta Software, P.O. Box 3522, Greenville, SC, 29608-3522

THE FLAGS OF THE 8K ROM PROGRAM

The ZX81 manual states that the three system variables FLAGS, FLAGX and CDFLAG hold various flags to control the BASIC system. But a study of the 8K ROM program discloses 12 separate flags.

FLAGS

Bit 0: This flag is reset whenever a space is to be printed before a token, and set if there is not to be a space. The reading of this flag takes place in the 'EXPAND TOKENS' subroutine at 094B. To see the action of the flag, enter: 10 REM 123456789 and POKE in the values 253, 203, 1, 134 or 198, 62, 255 or any other TOKEN code, 195, 75, 9. Execute the line by using RAND USR 16514.

Bit 1: This is the control flag for the printer. The flag is set for printing to the printer and reset for printing to the TV screen. The action of this flag is illustrated by altering the fourth byte of the Bit-0 program to 142 or 206. When 142 is used, the TOKEN prints on the TV display; when 206 is used the TOKEN passes to the printer and is not displayed.

Bit 2: This flag distinguishes between K and L modes. In 'CHOOS-ING K v L MODE' at 0537 the flag is reset for K and set for L mode.

Bit 6: This important flag distinguishes between the last value on the calculator stack being 'string' (therefore represented by parameters for its length and start) and 'numeric' (the value is held as a five byte floating-point number). The flag is reset for 'strings' and set for 'numbers.'

Bit 7: This is the syntax flag. Whenever a BASIC line is to have its syntax checked, the LINE-SCANNING subroutine is entered (at OCBA) and this flag is reset to indicate the need to check syntax. However, when a BASIC line is executed the routine is entered with bit 7 set. This is done at 0667 for each line.

FLAGX

Bit 0: This flag is used when the current value or string assigned to a given variable name has to be found in the variable area. The flag is reset when the search is for an array variable and set for simple variables.

Bit 1: This flag indicates if a current value or string was found or otherwise. It is reset if an 'old' value exists and becomes set if the variable name has not been used so far in the program.

Bit 5: This flag distinguishes between EDIT mode and INPUT mode. The 8K ROM program uses the same code to accept both types of input from the keyboard. The flag is reset for editing and becomes set when an INPUT command is being executed. The actual setting of the flag is done at OEF2, or OEF5 in the 'improved' ROM. (Some parts of the 'old' and 'new' 8K ROMs differ by 3 bytes--see David Ornstein's article, New 8K ROM Potpourri, Nov.81--AZ)

Bit 6: This flag distinguishes between numeric and string INPUTs. In the INPUT command routine this flag is made to be the same state as bit 6 of FLAGS and so is reset for a string and set for a number.

CDFLAG

This byte holds three flags and is perhaps the most interesting of the three flag system variables. The three flags all concern the ZX81's FAST/SLOW operation.

Bit 0: This flag remains reset until a frame of the display is stopped by pressing a key on the keyboard. The flag records this event by becoming set. In FAST mode this isn't very important, but in SLOW mode it results in the exiting from the 'SLOW display loop.' This loop occurs in the 8K

program at 04CF and comprises the two lines:

```
04CF SLOW-DISP BIT  0,(HL)      ;HL
                    ;points to CDFLAG
               JR Z,04CF,SLOW-DISP
```

This is a closed loop unless you realize that in SLOW mode the computer produces a 'frame' every 1/50th of a second. You can terminate a frame by pressing a key, and return from the interruption with Bit-0 of CDFLAG set.

Bit 6: This is the true FAST/SLOW flag. It is reset by executing the FAST command (0F23-0F26) and set by executing the SLOW command (0F28-0F2B). Resetting this bit is all that is required to change to FAST mode; however, the transition to SLOW mode is complicated by the need to also change bit 7.

Bit 7: This bit is usually a copy of bit 6. But certain tasks can only be performed in the FAST mode (like LOADing, SAVEing, and staying in PAUSE). When these tasks are to be performed then bit 7 is reset to act as a temporary marker for the need for FAST mode. The computer reads bit 7 as signalling which mode it should be in and not the true FAST/SLOW flag that only indi·cates the overall mode at the time.

Ian Logan, Skellingthorpe, England

Note: Further details are available in Ian's books, Understanding Your ZX81 ROM (for brief details) and The ZX81 Monitor Listing Part A, 0000-0F54 and Part B, 0F55-1DFE (for extensive details).--AZ

USER REPORT--INSIGHT'S 16K RAM PACK

Product: MX16-16K RAM
   From: PM Enterprises, 407 E. Lincolnway, Laporte, IN, 46530, 219/326-7923
   Price: $89.95 + $3 shipping
Note: Insight's RAM pack is now sold by PM Enterprises. It is the same unit.--AZ

At first glance you might ask, "Is that all there is to it?" I opened the shipping box to find a small black plastic case that obviously plugs into the rear of a ZX80/81 or MicroAce computer. The instructions were easy to understand and to the point. A simple program listing was also included. It tests the amount of RAM memory contained in the system to check for correct operation of the MX16.

The MX16 requires a different power supply to power the dynamic RAM chips inside. This 12-volt supply is included with the MX16 to replace the power supply that came with your computer. It plugs in the same way as your original supply. Your computer may run a little hotter than normal due to the increased voltage at the regulator. This should not be a problem. When I bought my MX16, Insight did not have any power supplies in stock. They offered to sell me an MX16 without the power supply at $10 off the regular price of $89.95. Also, Insight included a schematic for building a simple regulated power supply. Building your own may not be necessary--my standard ZX80 supply worked fine.

The only problem I encountered seemed to originate in the edge connector. Moving or bumping the computer even slightly caused a system crash. Cleaning the contacts on the computer PC board with and eraser and then a high quality electronic contact cleaner cleared up the problem.

I got my MX16 within 14 days of ordering. This should appeal to anyone waiting for Sinclair to ship their 16K RAM. Insight's technical support was very good--a letter or phone call brought fast results. PM Enterprises' service remains to be seen. The MX16 comes with a 90-day repair or replacement warranty.

I consider the MX16 to be an excellent value.

Lance M. Ward, Lansing, MI

MOVING BILLBOARD--8K/1K

This program continuously scrolls up to 23 (and more) lines of characters. Do not print more than 32 characters in any PRINT instruction, such as line 150, or the scrolling will fail.

These billboard lines are double-spaced. To increase the spacing, increase the difference between adjacent A(1) values the lines 70-74. If you use many PRINT lines, you must insert a CLS strategy. The use of DIM A(1) as a sum-to-memory strategy is discussed this page. Line 99 causes the billboard to repeat endlessly.

Paul Ezra, San Diego, CA

```
10 SCROLL
20 DIM A(1)
30 PRINT "MAJOR"
40 LET A(1)=A(1)+1
50 SCROLL
60 PAUSE (80)
70  IF A(1)=1 THEN GOTO 110
71  IF A(1)=2 THEN GOTO 140
72  IF A(1)=3 THEN GOTO 170
73  IF A(1)=4 THEN GOTO 200
74  IF A(1)=5 THEN GOTO 230
99  IF A(1)=6 THEN GOTO 10
100 GOTO 40
110 SCROLL
120 PRINT "BIRTHS:"
130 GOTO 40
140 SCROLL
150 PRINT "EARTH"
160 GOTO 40
170 SCROLL
180 PRINT "LIFE"
190 GOTO 40
200 SCROLL
210 PRINT "MAN"
220 GOTO 40
230 SCROLL
240 PRINT "CHRIST"
250 GOTO 40
```
Syntactic Sum: 28015, 8K

DIM AS CALCULATOR SUM-TO-MEMORY

DIM arrays can function as a sum-to-memory system found in many calculators. More importantly, it acts as a counter function and can be used where FOR-NEXT loops will not work. Moving Billboard, this page, illustrates the counter function. Here's a simple demonstration of the sum-to-memory system:

```
10 DIM B(1)
20 INPUT A
30 PRINT A
40 LET B(1)=B(1)+A
50 IF A=5 THEN PRINT B(1)
60 GOTO 20
```

Input a few numbers, hitting ENTER after each. Finish with 5 to execute line 50. B(1)'s value will appear and turns out to be the sum of the input numbers. To regain program listing, input any letter (not a number) and hit ENTER twice.

Change line 60 to 60 GOTO 10 and see that the sum-to-memory function is lost. Each time the computer executes a DIM line, all array values are cleared to zero.

Paul Ezra, San Diego, CA

8

SOLVING SIMULTANEOUS EQUATIONS

This 8K/1K program uses Crout's reduction to solve simultaneous equations. The number of variables you can use is limited only by your available memory; in 1K this program solves up to 4 equations in variables

When you type in the program, use the INPUT keyword for the input prompts in lines 2 and 25. For example, type 2 INPUT then back-space and hit PRINT. You'll get a space between " and INPUT, but it saves bytes. RUN the program, then enter the number of equations to solve in response to INPUT N

Enter the coefficients of the variables in matrix form. Say you're solving two equations in two variables: 4a+2b=17 and 2a+7b=10. In matrix form, your variables look like this:

	a	b	result
equ.1	4	2	17
equ.2	2	7	10

This is equivalent to this matrix:

	a	b	result
equ.1	K(1,1)	K(1,2)	K(1,3)
equ.2	K(2,1)	K(2,2)	K(2,3)

The first prompt asks for $K(1,1)$, so enter 4. $K(1,2)$ would be 2, $K(1,3)$ would be 17. Then $K(2,1)$ is 2, and so on.

The computer returns the answer in the form $U(1)$=(solution to first variable) and $U(2)$=(solution to second variable).

To find a polynomial of order N-1 to fit N points, add and change these lines:

```
12 PRINT " INPUT X";I
14 INPUT X
16 PRINT " INPUT Y";I
18 INPUT K(I,N+1)
19 CLS
```

delete lines 25, 35
change lines:

```
20 FOR J=1 TO N
30 LET K(I,J)=X**(J-1)
```

Enter the X and Y coordinates individually. N now equals the number of pairs of points. The computer returns the coefficients of the polynomial in this form:
$$Y=U(1)+U(2)X+U(3)X^2+U(4)X^3 \text{ etc.}$$
or $Y=U(Z)X^{(Z-1)}$.

Lori Olson, Quincy, MA

```
  2 PRINT " INPUT N"
  4 INPUT N
  6 CLS
  8 DIM K(N,N+1)
 10 FOR I=1 TO N
 20 FOR J=1 TO N+1
 25 PRINT " INPUT K(";I;",";J;")"
 30 INPUT K(I,J)
 35 CLS
 40 NEXT J
 50 NEXT I
 60 FOR I=1 TO N
 70 FOR J=I TO N
 80 FOR H=I TO I-1
 90 LET K(J,I)=K(J,I)-K(J,H)*K(H,I)
100 NEXT H
110 NEXT H
120 FOR J=I+1 TO N+1
130 FOR H=1 TO I-1
140 LET K(I,J)=K(I,J)-K(I,H)*K(H,J)
150 NEXT H
160 LET K(I,J)=K(I,J)/K(I,I)
170 NEXT J
180 NEXT I
190 FOR I=N TO 1 STEP -1
200 FOR H=I+1 TO N
210 LET K(I,N+1)=K(I,N+1)-K(I,H)*K(H,N+1)
220 NEXT H
230 NEXT I
240 FOR I=1 TO N
250 PRINT "U(";I;")=";K(I,N+1)
260 NEXT I
```

Syntactic Sum: 30176, 8K

TL$ TRANSLATIONS FOR 8K ROM

Substitute the lines on the right for lines on the left when translating 4K ZX80 programs to 8K:

	4K	8K
1.	PRINT TL$(STR$(M+A));	PRINT STR$(M+A))(2 TO);
2.	CODE (TL$(F$))	CODE F$(2 TO)
3.	CODE (TL$(TL$(F$))	CODE F$(3 TO)

DEAR EDITOR:

Many Sinclair personal computer owners will have noticed that Sinclair Research has changed its policy with regard to dealing with technical questions by phone. We have tried, and for a long time succeeded, in dealing with technical questions over the phone. However, a number of factors have made it impossible to continue providing this service.

First, we are selling more computers than ever before and it is simply not possible to recruit and train a sufficient number of qualified personnel to deal with technical inquiries.

Second, an increasing proportion of the computers that we sell are in kit form and the owners of these tend to have complex questions and more of them.

Third, most inquiries are not concerned with the operation of the computer, but rather with applications, modifications, etc.

Fourth, we try to make our products available at the lowest reasonable cost to the consumer and that simply does not allow us to engage in lengthy telephone conversations concerning, for example, possible applications of our products.

I would like to take this opportunity to make clear that our repair policies remain unchanged. If you have any problems with your computer within 90 days of purchase, it can be returned to us and will be repaired or replaced free of charge.

We will also try to answer any technical questions addressed to us in writing at 4 Sinclair Plaza, Nashua, NH, 03061. This service is exclusively for people who already own a Sinclair computer.

Nigel H. Searle, Sinclair Research, Boston, MA

SYNTAX will also answer technical questions by phone whenever we can. Call 617/456-3661 at a reasonable hour, Eastern Standard Time.--AZ

Having no knowledge WHATSOEVER of computers, I am considering the Sinclair ZX81 with 16K RAM. What I need is a simple Inventory Control system. Can you lead me to someone who has one?

Jerome Salny, Fairlawn, NJ

If anyone knows a source for such a program, please let me know.--AZ

I recently bought a ZX81 and am very interested in interfacing and control applications. Is anyone out there interested in implementing that exciting language, FORTH, on the ZX81?

Cesar A. Sepulveda, Pasadena, CA

Is there a way to hook up a big keyboard on the ZX81? Also why are there only 5 Sinclair software programs? I thought there would be a lot more.

Mark Biron, Manchester, NH

You can buy commercial keyboards to hook up. If you want to do it yourself, SYNTAX (Dec.80) published a plan to attach big keyboards to ZX80s. On a ZX80 you wire the keyswitches to diodes on the front of the PC board. A ZX81's keyboard is connected by ribbon cable to 2 headers, or connectors, (one 5-pin and one 8-pin). Where we showed connections to IC pins in our plan, just connect wires to a 5-pin connector. Where we showed diode connections, connect to an 8-pin connector. Then plug the new connectors onto the pins on the bottom of the ZX81 board.

According to Sinclair's office in Boston, Sinclair plans no more software. As to why so little, I suppose Sinclair chose to concentrate on hardware development.--AZ

I just received my ZX81 kit on Christmas eve and eagerly assembled it. While the instructions were well written, they deviate from the actual circuit board in 2 important areas. First, resistors R27 and R29 are mounted vertically near the modulator as marked on the PWB, not horizontally near the regulator as shown in the instructions Fig. 6. Second, the 2 small coils provided are not mentioned in the instructions or shown on the schematic. They connected the 9V dc input to the regulator when mounted at I1 and I2 on the PWB and the unit works fine--I only assume that was as intended!

Kit builders: Read the letter that accompanies the kit. It tells how to add four components that convert from UHF to VHF TV hookups.

Jay Walters, Shreveport, LA

Several ZX81 kit builders have found and resolved problems with their kits. If you're having troubles, call SYNTAX and we'll pass along their discoveries.--AZ

I recently bought a ZX81. As a novice, I am not pleased with the ZX81 BASIC Programming Manual. Is it possible to do regression (linear and multilinear) computations on this computer using a 16K memory module?

George A. Luhrs, Chula Vista, CA

In the Nov.81 issue Herb Sturges noted methods for improving ZX80 cooling. Mr. Sturges is in error about the improvement in heat transfer by painting the heat sink. I don't know how he derived the 20 times improvement over polished aluminum, but it is not true. Painting the heat sink can actually degrade the dissipation capability. A body's thermal blackness has ABSOLUTELY NO RELATION to its color in the visible spectrum. Of the three major means of heat transfer (radiation, conduction and convection), the blackness, or emissivity, affects only radiation. In most electronic applications other than spacecraft, conduction and convection are by far the dominant mechanisms.

Some years ago a company where I worked performed carefully controlled experiments confirming that there was usually no measurable difference in total dissipation between bare, clear anodized, black anodized, and painted aluminum.

The message: forget about the paint, cut the hole to improve convection, and ALWAYS use heat sink compound to improve conduction between the semiconductor and the heat sink. Unless you're going to orbit your ZX80, don't worry about radiative transfer. If you must have a black heat sink, paint it sparingly (so paint won't act as an insulator), masking off the area where the regulator mounts, and use lacquer, which has a slightly higher conductivity than enamel.

Gary J. Vincent, Richardson, TX

USERS' GROUPS

SEATTLE, WA, AREA: Contact Jeff Pack, 21026 109th SE #708, Kent, WA, 98031, 206/854-6232.

MICHIGAN: Lance Ward wants to organize a group with a monthly newsletter (dues to cover material and postage). Free ads to members and companies discounting products to members. Contact Lance at 433-D E. Edgewood Blvd., Lansing, MI, 48910, 517/393-1357.

CHICAGO, IL, AREA: L.P. Weigel is coordinator of the Chicago Area Computer Hobbyist Exchange Z80/Sinclair Special Interest Group. Yearly $10 dues include the CACHE newletter. For info, send a SASE c/o Box C-176, 323 S. Franklin #804. Chicago, IL, 60606, or call 312/561-6772.

RAM DECODING FOR ZX81

Last March we published Al Salt's plans for adding external RAM to ZX80s. To adapt his Build Additional RAM (SYNTAX 02/03/02) with ZX81s, you can take one of three approaches. The second two of these plans do not require you to modify the insides of your ZX81.

You can use all external RAM, by removing the internal RAM and connecting pins 2 and 14 of the 74155 to pin 2T (for top) of the ZX81 edge connector. This method uses pin 2T as an output from the computer to the decoder. Connect CS1 to the first set of external RAM chips.

Or, wire a 74LS138 decoder wired as shown in the following circuit to select the internal set or one of seven external sets of RAM chips. Use pin 2T as an input to the computer from the external

decoder in this case. (Sinclair used this scheme in their 3K RAM module sold in Europe. Henry April of the Boston Computer Society independently proposed a similar plan at a recent Society meeting.)

Finally, you can add a 74LS-series inverter (74LS04, for example) in series with A14 and connect the inverted output to pin 2 of the 74155 decoder. (See additional schematic following.) Then connect MREQ to pin 14 of the 74155. Connect the RAM CS1 form pin 9 of the decoder to pin 2T of the ZX81. This scheme uses pin 2T as an input to the computer from the external decoder. (This plan comes from John Jarrett's article for SYNTAX on RAM expansion, now being tested.)

Each of the latter schemes is logically correct and used in one or more memory expanders, but we did NOT bench test them.
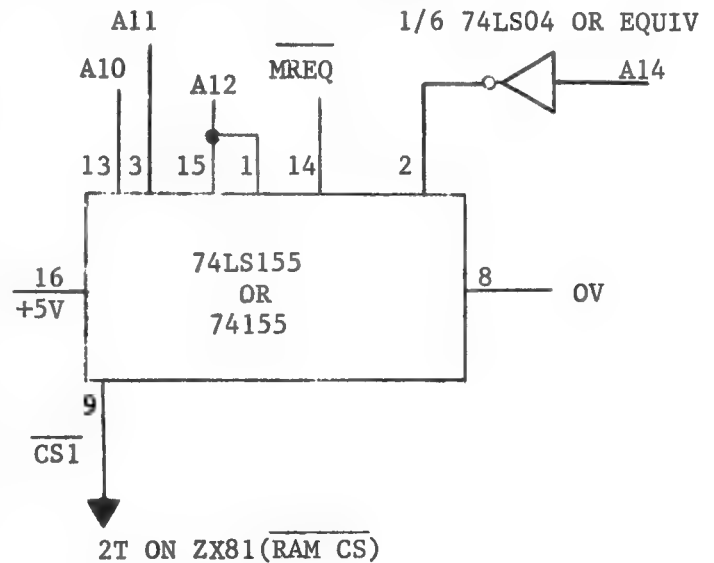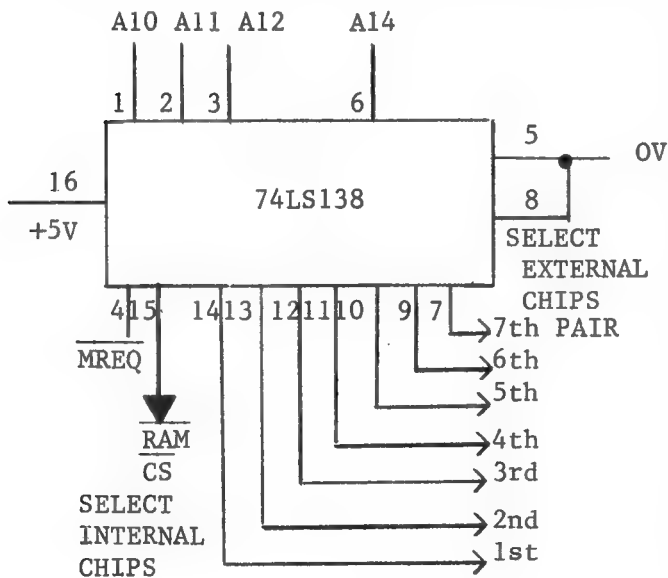
LOADING PROBLEMS WITH ZX81 KITS

If you have loading problems with a kit ZX81, see if you cracked C10 during assembly. Some machines LOAD fine with 1 (or 2)K internal memory, but not with 16K. If you hear "shortwave radio noises" (high pitched sounds) with 16K attached, try this. Move C11 and R27 to be physically attached at the MIC connector of your ZX81. This also works on assembled ZX81s, if you have the skill to do it, but voids your warranty.

NUCLEAR DECAY SIMULATION--4K/1K

A popular lab exercise uses special dice to simulate radioactive decay. Marked die faces show the decay probability. After shaking and throwing the dice, replace those with marked sides showing with spheres (stable atoms) or dice with different decay probability (radioactive daughters). Use over 200 atoms for a smooth graph.

This program throws, examines and exchanges dice. You get data to plot a decay chain from radioactive parent to stable population.

The prompt asks for the number of generations. Enter 2 to decay to a stable isotope, or 3 to 5 for a decay chain. Next, enter the relative half-life of each generation, corresponding to the total number of faces on the die. For example, 8 denotes a 1 in 8 chance of decay for each atom in that generation during each pass. Then, enter the initial number of atoms. The program stops when the population consists entirely of stable atoms. Graphs derived from this program agree well with experiment.

S. Sanderson, Carpentersville, IL

```
10 DIM H(5)
20 DIM A(5)
30 PRINT "NUCLEAR DECAY SIMULA
TION"
40 PRINT
50 PRINT
60 PRINT "HOW MANY GENERATIONS
? (>1)"
70 INPUT G
80 PRINT
90 PRINT "RELATIVE 1/2 LIFE (F
ACES ON DIE)"
100 FOR I=1 TO G-1
110 PRINT "GEN. (";I;") = ";
120 INPUT H(I)
130 PRINT H(I)
140 NEXT I
150 LET A(G)=0
160 LET K=0
170 PRINT "INITIAL NO. OF ATOMS
?"
180 INPUT A
190 LET A(1)=A
200 FOR I=1 TO G-1
210 IF A(I)=0 THEN GO TO 280
220 LET N=A(I)
230 FOR J=1 TO N
240 IF NOT RND(H(I))=1 THEN GO
TO 270
250 LET A(I)=A(I)-1
260 LET A(I+1)=A(I+1)+1
270 NEXT J
280 NEXT I
290 LET K=K+1
300 CLS
310 PRINT "PASS ";K
320 PRINT
330 FOR I=1 TO G
340 PRINT "GEN. (";I;") = ";A(I
)
350 NEXT I
360 IF A(G)=A THEN GO TO 410
370 PRINT "NL TO CONTINUE"
380 INPUT A$
390 IF A$="" THEN GO TO 200
400 STOP
410 PRINT
420 PRINT "ALL ATOMS STABLE"
```
Syntactic Sum: -17745, 4K

SALARY SCHEDULING COST COMPUTATION

This 8K/16K program computes salary schedules for two or three classes of teachers (or other workers). For a two column schedule, enter all zeros for the prompts about the third class (Mast+). At the end of the second column, it will ask if you want another. Enter NO and touch NL. The computer will give you schedule cost and total number of teachers for the entire two column schedule. For a three column display, answer YES to the prompt and enter data for the third class.

For both two- and three-column schedules, you see column cost, number of teachers and number of increments at the end of each column.

Lines 65, 67 and 69 (inputs BY, MY, MAY) determine the number of years in each column. Lines 41-63 are prompts and inputs for the initial salary in each column and the increment size for each column. Increments within a column must be the same, but they may vary between columns. Line 19 is an input statement for the school/corporation name.

Each column's cost is calculated: ed. cost=(#teachers* beginning sal.)+(#increments*increment size). Knowing the number of increments in each column is very useful in schedule development.

Robert M. DeMunbrun, Rushville, IN

```
   1 PRINT TAB 8;"EQUAL INCREMEN
T"
   3 PRINT TAB 8;"SALARY SCHEDUL
ING"
   6 PRINT TAB 7;"COST COMPUTATI
ON"
   9 PRINT TAB 15;"BY"
  12 PRINT TAB 7;"R.M. DEMUNBRUN
"
  15 PRINT
  18 PRINT "SCHOOL CORP NAME?"
  19 INPUT A$
  20 PRINT A$
  21 PRINT
  24 PRINT TAB 6;"1980 81 SCHOOL
YEAR"
  27 PRINT
  36 PRINT
  39 PRINT
  41 PRINT "BEG BACH SALARY?"
  43 INPUT B
  45 PRINT "BACH INCREMENT?"
  47 INPUT I1
  49 PRINT "BEG MAST SALARY?"
  51 INPUT M
  53 PRINT "MAST INCREMENT?"
  55 INPUT I2
  57 PRINT "MAST+ SALARY?"
  59 INPUT MA
  61 PRINT "MAST+ INCREMENT?"
  63 INPUT I3
  64 PRINT "YRS BACH COLUMN?"
  65 INPUT BY
  66 PRINT "YRS MAST COLUMN?"
  67 INPUT MY
  68 PRINT "YRS MAST+ COLUMN?"
  69 INPUT MAY
  70 LET N=-1
  72 PRINT TAB 1;"DEGREE";TAB 17
;"YEARS"
  73 PRINT TAB 1;"TEACHERS";TAB
17;"INCREMENTS"
  74 PRINT
  75 LET TE1=0
  95 LET IN1=0
 100 LET N=N+1
 105 IF N>BY THEN GOTO 190
 110 PRINT TAB 1;"BACH";TAB 17;N
 120 INPUT NO
 130 LET TE1=TE1+NO
 140 LET IN=N*NO
 150 LET IN1=IN1+IN
 170 PRINT TAB 1;NO;TAB 17;IN
 180 PRINT
 185 GOTO 100
 190 LET C=B*TE1
 200 LET CIN=IN1*I1
 210 LET COST1=C+CIN
 220 PRINT TAB 1;"BACH COST";TAB
16;"$";TAB 17;COST1
 230 PRINT
 240 PRINT TAB 1;"NO. TEACH";TAB
17;TE1
 250 PRINT
 260 PRINT TAB 1;"NO. INCREMENTS
";TAB 17;IN1
 280 PRINT
 290 LET N=-1
```

```
300 LET TE2=0
330 LET IN2=0
340 LET N=N+1
350 IF N>MY THEN GOTO 430
360 PRINT TAB 1;"MAST";TAB 17;N
370 INPUT NO
380 LET TE2=TE2+NO
390 LET IN=N*NO
400 LET IN2=IN2+IN
410 PRINT TAB 1;NO;TAB 17;IN
420 PRINT
425 GOTO 340
430 LET C2=M*TE2
440 LET CIN2=IN2*I2
450 LET COST2=C2+CIN2
460 PRINT TAB 1;"MAST COST";TAB
16;"$";TAB 17;COST2
470 PRINT
480 PRINT TAB 1;"NO. TEACH";TAB
17;TE2
490 PRINT
500 PRINT TAB 1;"NO. INCREMENTS
";TAB 17;IN2
510 PRINT
520 PRINT "ANOTHER COLUMN? ""YE
S"" OR ""NO"""
530 INPUT B$
540 IF B$="YES" THEN GOTO 600
550 LET TC1=COST1+COST2
560 PRINT TAB 1;"SCHEDULE COST"
;TAB 16;"$";TAB 17;TC1
580 LET TEA=TE1+TE2
590 PRINT TAB 1;"NO. TEACHERS";
TAB 17;TEA
595 GOTO 900
600 LET N=-1
610 LET TE3=0
640 LET IN3=0
645 LET C=0
650 LET N=N+1
660 IF N>MAY THEN GOTO 740
670 PRINT TAB 1;"MAST+";TAB 17;
N
680 INPUT NO
690 LET TE3=TE3+NO
700 LET IN=N*NO
710 LET IN3=IN3+IN
720 PRINT TAB 1;NO;TAB 17;IN
730 PRINT
735 GOTO 650
740 LET C=MA*TE3
750 LET CIN3=IN3*I3
760 LET COST3=C+CIN3
770 PRINT TAB 1;"MAST+ COST";TA
B 16;"$";TAB 17;COST3
780 PRINT
790 PRINT TAB 1;"NO. TEACH";TAB
17;TE3
800 PRINT
810 PRINT TAB 1;"NO. INCREMENTS
";TAB 17;IN3
820 PRINT
825 PRINT
830 LET TEACH=TE1+TE2+TE3
840 PRINT TAB 1;"TOT. TEACH";TA
B 17;TEACH
850 PRINT
860 LET TC2=COST1+COST2+COST3
870 PRINT TAB 1;"SCHEDULE COST"
;TAB 16;"$";TAB 17;TC2
900 STOP
```
Syntactic Sum: 64742, 8K

BOOK REVIEW

Title: Sinclair ZX81 ROM
         Disassembly Part A:
         0000H-0F54H
   By: Dr. Ian Logan
 From: Melbourne House Ltd.
Price: $20.00 in US

   Ian Logan has disassembled and
annotated the ZX81 Monitor, easing
your access to the routines of the
improved 8K ROM. (If you have an
old ROM--PEEK 54=132--this disas-
sembly doesn't match your machine.)
This analysis is more up-to-date
than the 1st ed. of The ZX81
Companion. Ian's book includes all
functions except for the floating
point calculator.
   I find the book useful and
reasonably done, although several
obvious typos make me nervous (p.2:
0028 listed twice, the second is
002B; p.29: 16417 equals 4021H,
not 401FH). I would verify any
routine I intended to use by
PEEKing appropriate locations.
   Preceding every major routine,
you will find a concise functional
description of the code--very
useful for quick understanding.
   In particular, you can examine
the LPRINT, LOAD and SAVE routines.
These last two differ from the 4K
routines published earlier (SYNTAX,
May 81 & Jul.81).

Logan's book doesn't list hex codes following addresses, which makes checking difficult. Also, the notation differs from Logan's earlier 4K ROM disassembly; here you'll find both subroutine names and hex addresses in commands (like CALL 031E,OUT-BYTE). Compare the CALL example with LD HL,+VERSN (see sample routine following). Although good to locate subroutines, the names aggravate the problem of counting bytes. If you count bytes between 030B and 031E, only one name (VERSN) has bytes associated with it. Further, the typeface doesn't distinguish system variables carrying Sinclair names, from those which Dr. Logan named.

Logan's notes on the system variables include both decimal and hex addresses, but not the save and poke codes from Sinclair's manual. Logan details the flag functions by bit (as Sinclair did not), and his alternative expressions of system variables enhances understanding.

All things considered, Logan's comments plus the additional data on flags add real value to the disassembly. In future editions, I would urge Dr. Logan to include each address and the code in the ROM as well as to separate variable names from the commands. Also, system variables should stand out from monitor calls.--KO

THE 'SAVE' COMMAND ROUTINE

HL is set to point to the start of the program name. There is a 6 second header and then the bytes of the name and the program are passed out to the cassette recorder.

02F6 SAVE	CALL	03A8,NAME
	JR	C,02F4,REPORT-F
	EX	DE,HL
	LD	DE,+12CB
02FF HEADER	CALL	0F46,BREAK-1
	JR	NC,0332,BREAK-2
0304 DELAY-1	DJNZ	0304,DELAY-1
	DEC	DE
	LD	A,D
	OR	E
	JR	NZ,02FF,HEADER
030B OUT-NAME	CALL	031E,OUT-BYTE
	BIT	7,(HL)
	INC	HL
	JR	Z,030B,OUT-NAME
	LD	HL,+VERSN
0316 OUT-PROG.	CALL	031E,OUT-BYTE
	CALL	01FC,LOAD/SAVE
	JR	0316,OUT-PROG.
031E OUT-BYTE	LD	E,(HL)

8K MEMORY PEEK

This slow but simple program peeks memory 100 bytes at a time. It displays consecutive bytes in 5 columns by 20 rows. Designed for the ZX81, it works with 16K RAM.

To use, enter the program, then RUN. Type any address starting with an even hundred, then ENTER (use 0 to see the first 100 bytes in ROM). After 15 seconds the screen displays 100 bytes starting at the address you entered. Type CONT ENTER and the next screen listing continues at the next address. The program runs faster in the FAST mode on a ZX81.

We can provide this program on cassette for $7.31 if you want a working copy. We cannot troubleshoot your typed program.

Joseph L. Hartmann Jr., 101 Tonnele Ave, Jersey City, NJ 07306

```
  10 PRINT "ADDRESS IS =",
  20 INPUT A
  30 CLS
  35 PRINT AT 0,6;"WAIT 15 SECON
DS"
  37 PAUSE 80
  38 POKE 16437,255
  40 PRINT AT 0,6;"+00 +20 +40 +
60 +80"
  50 FOR I=1 TO 20
  60 PRINT AT I+1,0;A-1+I
  70 NEXT I
  90 FOR J=1 TO 5
  95 FOR I=1 TO 20
 100 PRINT AT I+1,6+4*(J-1);PEEK
(A-1+I)
 110 NEXT I
 120 LET A=A+20
 130 NEXT J
 140 STOP
 150 GOTO 40
```
Syntactic Sum: 19776, 8K

Readers have written in looking for users' groups in San Francisco, San Diego, and Manhattan, NY. Send us information about your group and we'll put interested people in touch with you.

AWARI--8K/16K

     Awari is an ancient African game for two players. Six piles of three stones each are arranged in front of opposing players, with each player's "home" to the right of the piles. The play consists of taking all the stones in one of the player's piles and, moving counter-clockwise, dropping one stone in each pile or home until all are dropped. If a player's last stone falls in his home, then he gets a second move. If the last stone falls in an empty pile and the opposite pile is not empty, then the player can move the last stone plus all stones in the opposite pile to his home. The game ends when either player runs out of moves. The player with the most stones in his home wins.

Sample strategy:

```
      5   0   1   3   0   4
  5                           5
      3   3   4   0   2   1
```

The computer would most likely move its number 6 pile (top left with 5 counters), since it would end up with 1+1+3 or 5 more counters in its home. Moving its pile 1 or 4 would only result in 4 more points. Were it your turn, a good defensive move would be your pile 2 with 3 counters. But moving pile 3, 5 or 6 first would be even better since you would get 1 more point plus a second move to take pile 2 out of harm's way.

    This Awari game is a simpli-fied version of one in Basic Compu-ter Games by Creative Computing. Although it only knows offensive moves, it plays a reasonably good game. If you're like me, you'll lose often enough to keep it inter-esting. The first player to move is chosen randomly, and then the first move alternates if you press NL or ENTER at the end of each game. Input STOP for your move to quit, then continue with GOTO 940 or RUN. Input any character at the end of a game to see the listing.

    Note that the <> (not equal to) sign is shift T; don't type less than and greater than signs. To enter lines 690 and 710, type the line number, PRINT command and " mark, then press shift 9 to change to graphics. Press shift 7 25 times for line 690, shift 5 once for line 710. Press shift 9 again to return to normal input mode, then type the ending " mark.

Jon Passler, Beverly, MA

```
10 REM AWARI
20 RAND
30 LET K1=INT (RND*2)+1
40 DIM A(14)
50 DIM S(14)
60 FOR I=1 TO 13
70 LET A(I)=3
80 NEXT I
90 LET A(7)=0
100 LET K=K1
110 LET Z=3
120 LET J=15
130 GOSUB 68C
140 GOTO K*40+110
150 PRINT "YOUR MOVE";
160 GOTO 230
170 PRINT "AGAIN";
180 GOTO 230
190 PRINT "MY TURN-N/L";
195 INPUT I$
200 GOTO 270
210 PRINT "AGAIN-N/L";
215 INPUT I$
220 GOTO 270
230 INPUT M
240 IF A(M)=0 THEN GOTO 230
250 PRINT ": ";M
260 GOTO 500
270 IF Z>3 THEN GOTO 300
280 LET M=INT (RND*2)+12
290 GOTO 490
300 FOR I=1 TO 14
310 LET S(I)=A(I)
320 NEXT I
330 LET H1=A(14)-A(7)
340 FOR J=8 TO 13
350 IF A(J)=0 THEN GOTO 470
360 LET M=J
```

17

```
370 GOSUB 500
380 LET H2=A(14)-A(7)
390 IF M=14 AND K=2 THEN LET H2
=H2+2
400 IF ABS (H1-H2)<2 THEN LET H
2=H2+2
410 IF H1>H2 THEN GOTO 440
420 LET H1=H2
430 LET W=J
440 FOR I=1 TO 14
450 LET A(I)=S(I)
460 NEXT I
470 NEXT J
480 LET M=W
490 PRINT ": ";M-7
500 LET I=A(M)
510 LET A(M)=0
520 FOR I=I TO 1 STEP -1
530 LET M=M+1
540 IF M>14 THEN LET M=M-14
550 LET A(M)=A(M)+1
560 NEXT I
570 IF A(M)=1 AND M<>7 AND M<>1
4 THEN IF A(14-M)<>0 THEN GOTO 5
90
580 GOTO 620
590 LET A(INT K*7)=A(INT K*7)+A
```

```
(14-M)+1
600 LET A(14-M)=0
610 LET A(M)=0
620 IF J<14 THEN RETURN
630 IF Z=3 THEN GOTO 670
640 FOR I=1 TO 11
650 SCROLL
660 NEXT I
670 LET Z=14
680 PRINT AT 0,12;"AWARI"
690 PRINT AT Z-2,1;"▬▬▬▬▬▬
▬▬▬▬▬▬▬" (25 shift 7)
700 FOR I=Z-2 TO Z+7
710 PRINT AT I,26;"█" (shift 5)
720 NEXT I
730 PRINT AT Z,5;A(13);"  ";A(1
2);"  ";A(11);"  ";A(10);"  ";A(
9);"  ";A(8)
740 PRINT AT Z+2,1;"";
750 IF A(14)<10 THEN PRINT " ";
760 PRINT A(14);TAB 23;A(7)
770 PRINT AT Z+4,5;A(1);"  ";A(
2);"  ";A(3);"  ";A(4);"  ";A(5)
;"  ";A(6)
780 PRINT AT Z+7,1;"";
790 IF Z=3 THEN RETURN
800 FOR I=1 TO 6
810 IF A(I)<>0 THEN GOTO 840
820 NEXT I
830 GOTO 870
840 FOR I=8 TO 13
850 IF A(I)<>0 THEN GOTO 980
860 NEXT I
870 LET D=A(14)-A(7)
880 IF D=0 THEN PRINT "DRAW"
890 IF D<0 THEN PRINT "YOU WIN
BY ";-D
900 IF D>0 THEN PRINT "I WIN BY
";D
910 INPUT I$
920 CLS
930 IF I$<>"" THEN LIST 10
940 IF K1=1 THEN LET W=2
950 IF K1=2 THEN LET W=1
960 LET K1=W
970 GOTO 40
980 IF (M=7 AND K=1) OR (M=14 A
ND K=2) THEN GOTO 1030
990 LET W=K
1000 IF W<2 THEN LET K=2
1010 IF W>1.5 THEN LET K=1
1020 GOTO 140
1030 LET K=K+.5
1040 GOTO 140
Syntactic Sum:  5918, 8K
```

CLASSIFIED ADS

With SYNTAX classifieds, you can reach thousands of ZX80, ZX81 and MicroAce users to sell, swap or inform. Just $2.75 per line, 4-line min. Send your ad, typed 35 characters per line, with check or credit card no. by 15th of any month for following month's issue to SYNTAX, RD 2 Box 457, Harvard, MA 01451. No phone orders, please.

OUR POLICY ON CONTRIBUTED MATERIAL

SYNTAX ZX80 invites you to express opinions related to the ZX80 and the newsletter. We will print, as space allows, letters discussing items of general interest. Of course, we reserve the right to edit letters to a suitable length and to refuse publication of any material.

We welcome program listings for all levels of expertise. Programs can be for any fun or useful purpose. We will test run each one before publishing it, but we will not debug programs; please send only workable listings.

In return for your listing, we will pay you a token fee of $2.00 per program we use. This payment gives us the nonexclusive right to use that program in any form, world-wide. This means you can still use it, sell it, or give it away, and so can we.

We will consider submissions of news and hardware or software reviews. Please keep articles short (350-400 words). Again, we reserve the right to edit accepted articles to a suitable length. We will pay 7 cents per 6 characters, including spaces and punctuation, for accepted articles.

When you send in programs for possible publication in SYNTAX, please include the following information:
- How to operate the program, including what to input if it does not contain prompts.
- Whether you can run the program over again and how.
- How to exit the program.
- The Syntactic Sum (using the Syntactic Sum program in the February, 1981, issue).
- Whether it fits in 1K or 2K RAM (or 16K when available).
- Whether it uses the 4K or 8K ROM.

We pay for this explanatory text at the same rate as for articles in addition to payment for the program itself.

If you want us to return your original program listing or article, please include a self-addressed, stamped envelope. Otherwise, we cannot return submitted material.

D&D fans love Solo Dungeons with ZX80-81 state which.Listing $5.0 Sealed letters for special rooms Halfling 454 Hallsville TX 75650

4 HOME FINANCE PROGRAMS ON CASSETTE Income Averaging;Mortgage Payments; Auto Loan;Interest Analysis;(8K/1K) All for $5.95;Ck payable to J.Pyryt 95 Union Avenue, Garfield, NJ 07026

ZX81 SOFTWARE - For 1K or 16K RAM. Games, business,math, and more! Send SASE for free catalog to: Great Lakes Software,201 Burlington Rd.,Valparaiso, IN 46383

#6116 2K x 8 150 NS Static Ram Chip Available for delivery @ $19.00 plus $1.00 for shipping. Send check or MO to: Support Systems One, PO Box 1794, Phila., PA 19105

7 1K ZX81 GAMES ON CASSETTE--$10.00 Mastermind, Slot Machine, Craps, Sub Rescue, White Hot Number, and more. NEW ENGLAND SOFTWARE,Box 691, Hyannis, MA 02601. (Also for ZX80).

New 57 key keyboards with keytops similar to ZX80 and ZX81 words on

top. All commands available on keys
Prewired $55.00 kit $45.00 without
printed keytops $35.00 kit-$25.00
Kit for putting on keytops $10.00
Z80A PIO chip $4.10 Free catalog.
Calif. res. add tax. Incl. $2.50
for shipp. & hand. A.&P. Elec-
tronics 5945 Kenneth Ave #34
Carm. CA. 95608

ZX81/1K Listings: imp UFO keeps
score, 3 ships. Bomb Search, 9
stories 81 rooms. $1+SASE Fr. Bowes
Box 148 Hopewell Jct. NY 12533.

Wanted to buy--back issues of Sync
or SYNTAX magazines. J.S. Woodson
K4DP 3154 Pitzer's Ridge Rd.
Covington, VA. 24426 (703)962-4268

FOR SALE: Sinclair ZX81 with Sin-
clair 16K Memory Module and manual.
Purchased Nov. '81-$200.00
Call- (617) 668-7896 aft. 7PM

16K GAMES FOR THE ZX81/ZX80 8K ROM
ZX8 TREK - 5 levels of difficulty
  with 20 battle command options.
CRIBBAGE - the computer challenges
  you at the game of Cribbage.
TANK BATTLE - find the computer's
  tank before it locates yours.
Cassette + manual - $14.95 ea. game
A.Nisbet, 6 Moffatt Court, Rexdale,
    Ontario, Canada, M9V 4E1